THE CLAIMS

The following listing of claims replaces all previous listings of the claims pending in the application.

1. (Currently Amended) ~~An apparatus~~ A computer for performing correctness checks opportunistically, the ~~apparatus~~ computer comprising:

first logic, the first logic receiving a first set of instructions and generating an initial instruction schedule from the first set of instructions, the first set of instructions including one or more instructions associated with a correctness check function;

second logic, the second logic evaluating the initial instruction schedule to determine whether the initial instruction schedule includes spare instruction slots into which said one or more instructions associated with the correctness check function can be inserted; ~~and~~

third logic, the third logic determining a number of additional instruction slots that may be added to the initial instruction schedule without exceeding a run-time performance cost tolerance level; and

fourth logic, the fourth logic inserting said one or more instructions associated with the correctness check function into the spare instruction slots when enough spare instruction slots exist in the initial instruction schedule for accommodating said one or more instructions in a final schedule of instructions, wherein when enough spare instruction slots do not exist in the initial instruction schedule to accommodate said one or more instructions, the fourth logic determines whether the number of additional instruction slots is large enough to accommodate said one or more instructions, when the fourth logic determines that the number of additional instruction slots is large enough to accommodate said one or more instructions, the fourth logic inserts said one or more instructions into the additional instruction slots.

2. (Currently Amended) The ~~apparatus~~ computer of claim 1, ~~further comprising:~~

~~fourth logic, the fourth logic inserting said one or more instructions associated with the correctness check function into the spare instruction slots if enough spare instruction slots exist in the initial instruction schedule for accommodating said one or more instructions~~ wherein the fourth logic inserts said one or more instructions in

spare instruction slots located within a range of instructions where dependencies are satisfied.

3. (Currently Amended) The ~~apparatus~~ computer of claim ~~2~~ 1, wherein ~~if enough spare instruction slots do not exist in the initial instruction schedule to accommodate said one or more instructions, the fourth logic determines whether the number of additional instruction slots is large enough to accommodate said one or more instructions, wherein if the fourth logic determines that the number of additional instruction slots is large enough to accommodate said one or more instructions,~~ the fourth logic inserts said one or more instructions into the additional instruction slots located within a range of instructions where dependencies are satisfied.

4. (Currently Amended) The ~~apparatus~~ computer of claim 1, wherein said one or more instructions associated with the correctness check function correspond to a conditional expression, and wherein the first logic performs initial code generation prior to generating the initial instruction schedule, wherein when the first logic performs initial code generation, said one or more instructions associated with the correctness check function are separated from all other instructions of said first set of instructions so that the initial instruction schedule does not include any instructions associated with the correctness check function.

5. (Currently Amended) The ~~apparatus~~ computer of claim 1, wherein said first, second and third logic correspond to a processor programmed to execute a compiler program, the compiler program including a first code segment for performing initial code generation and for generating the initial instruction schedule, a second code segment for evaluating the initial instruction schedule to determine whether spare instruction slots exist in the initial instruction schedule, and a third code segment for determining a number of additional instruction slots that may be added to the initial instruction schedule without exceeding a run-time performance cost tolerance level.

6. (Currently Amended) The ~~apparatus~~ computer of claim ~~2~~ 1, wherein said first, second, third and fourth logic correspond to a processor programmed to execute a compiler program, the compiler program including a first code segment that causes the processor to perform initial code generation and to generate the initial instruction schedule, a second code segment that causes the processor to evaluate the initial instruction schedule to determine whether spare instruction slots exist in the initial instruction schedule, a third code segment that causes the processor to determine the number of additional instruction slots that may be added to the initial instruction schedule without exceeding a run-time performance cost tolerance level, and a fourth code segment that causes the processor to insert said one or more instructions into the spare instruction slots if enough spare instruction slots exist in the initial instruction schedule to accommodate said one or more instructions.

7. (Currently Amended) The ~~apparatus~~ computer of claim 6, wherein if enough spare instruction slots do not exist in the initial instruction schedule to accommodate said one or more instructions, the fourth code segment causes the processor to determine whether the number of additional instruction slots is large enough to accommodate said one or more instructions, wherein if the processor determines that the number of additional instruction slots is large enough to accommodate said one or more instructions, the processor inserts said one or more instructions into the additional instruction slots.

8. (Currently Amended) A method for performing correctness checks opportunistically performed in an executable instruction processing system, the method comprising ~~the steps of~~:

receiving a first set of instructions and generating an initial instruction schedule from the first set of instructions, the first set of instructions including one or more instructions associated with a correctness check function;

evaluating the initial instruction schedule to determine whether the initial instruction schedule includes spare instruction slots into which said one or more instructions associated with the correctness check function can be inserted; ~~and~~

determining a number of additional instruction slots that may be added to the initial instruction schedule without exceeding a run-time performance cost tolerance level; and

inserting said one or more instructions associated with the correctness check function into the spare instruction slots if enough spare instruction slots exist in the initial instruction schedule for accommodating said one or more instructions to create a final schedule of instructions, wherein when enough spare instruction slots do not exist in the initial instruction schedule to accommodate said one or more instructions,

determining whether the number of additional instruction slots is large enough to accommodate said one or more instructions; and when a determination is made that the number of additional instruction slots is large enough to accommodate said one or more instructions,

inserting said one or more instructions into the additional instruction slots.

9. (Currently Amended) The method of claim 8, further comprising ~~the steps of~~:

~~inserting said one or more instructions associated with the correctness check function into the spare instruction slots if enough spare instruction slots exist in the initial instruction schedule for accommodating said one or more instructions~~ inserting said one or more instructions in spare instruction slots located within a range of instructions where dependencies are satisfied.

10. (Currently Amended) The method of claim ~~9~~ 8, ~~further comprising~~ ~~the step of~~:

~~if enough spare instruction slots do not exist in the initial instruction schedule to accommodate said one or more instructions, determining whether the number of additional instruction slots is large enough to accommodate said one or more instructions; and~~

~~if a determination is made that the number of additional instruction slots is large enough to accommodate said one or more instructions, inserting said one or more instructions into the additional instruction slots~~ wherein inserting said one or more instructions into the additional instruction slots comprises inserting said one or more instructions within a range of instructions where dependencies are satisfied.

11. (Currently Amended) The method of claim ~~10~~ 8, wherein said one or more instructions associated with the correctness check function correspond to a conditional expression, and wherein the step of performing initial code generation is performed prior to generating the initial instruction schedule, wherein when initial code generation is performed, said one or more instructions associated with the correctness check function are separated from all other instructions of said first set of instructions so that the initial instruction schedule does not include any instructions associated with the correctness check function.

12. (Original) The method of claim 8, wherein the method is performed by a processor programmed to execute a compiler program, the compiler program including a first code segment for performing initial code generation and for generating the initial instruction schedule, a second code segment for evaluating the initial instruction schedule to determine whether spare instruction slots exist in the initial instruction schedule, and a third code segment for determining a number of additional instruction slots that may be added to the initial instruction schedule without exceeding a run-time performance cost tolerance level.

13. (Currently Amended) A computer program for performing correctness checks opportunistically, the computer program ~~being~~ embodied on a computer-readable medium, the computer program comprising:

a first code segment~~, the first code segment generating~~ that when executed generates an initial instruction schedule from a first set of instructions, the first set of instructions including one or more instructions associated with a correctness check function;

a second code segment~~, the second code segment evaluating~~ that when executed evaluates the initial instruction schedule to determine whether the initial instruction schedule includes spare instruction slots into which said one or more instructions associated with the correctness check function can be inserted; ~~and~~

a third code segment~~, the third code segment determining~~ that when executed determines a number of additional instruction slots that may be added to the initial instruction schedule without exceeding a run-time performance cost tolerance level;

a fourth code segment that when executed inserts said one or more instructions associated with the correctness check function into the spare instruction slots when enough spare instruction slots exist in the initial instruction schedule to accommodate said one or more instructions in a final schedule of instructions; and

a fifth code segment, wherein if the fourth code segment determines that enough spare instruction slots do not exist in the initial instruction schedule to accommodate said one or more instructions, the fifth code segment when executed determines whether the number of additional instruction slots is large enough to accommodate said one or more instructions, and wherein when a determination is made that the number of additional instruction slots is large enough to accommodate said one or more instructions, the fifth code segment causes said one or more instructions to be inserted into the additional instruction slots of the final schedule.

14. (Currently Amended) The computer program of claim 13, ~~further comprising:~~

~~a fourth code segment that when executed inserts said one or more instructions associated with the correctness check function into the spare instruction slots when enough spare instruction slots exist in the initial instruction schedule to accommodate said one or more instructions~~ wherein said one or more instructions are inserted in spare instruction slots located within a range of instructions where dependencies are satisfied.

15. (Currently Amended) The computer program of claim ~~14~~ 13, ~~further comprising:~~

~~a fifth code segment, wherein if the fourth code segment determines that enough spare instruction slots do not exist in the initial instruction schedule to accommodate said one or more instructions, the fifth code segment determines whether the number of additional instruction slots is large enough to accommodate said one or more instructions, and wherein if a determination is made that the number of additional instruction slots is large enough to accommodate said one or more instructions, the fifth code segment causes said one or more instructions to be inserted into the additional instruction slots~~ wherein said one or more instructions are inserted

in the additional instruction slots located within a range of instructions where dependencies are satisfied.

16. (Currently Amended) The computer program of claim ~~15~~ 13, wherein said one or more instructions associated with the correctness check function correspond to a conditional expression, and wherein ~~the step of~~ performing initial code generation is performed prior to generating the initial instruction schedule, wherein when initial code generation is performed, said one or more instructions associated with the correctness check function are separated from all other instructions of said first set of instructions so that the initial instruction schedule does not include any instructions associated with the correctness check function.